UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/601,444	06/23/2003	Michael L. Brundage	MSFT-1753/301638.1	7697

41505 7590 02/07/2008

WOODCOCK WASHBURN LLP (MICROSOFT CORPORATION)
CIRA CENTRE, 12TH FLOOR
2929 ARCH STREET
PHILADELPHIA, PA 19104-2891

EXAMINER
GORTAYO, DANGELINO N

ART UNIT	PAPER NUMBER
2168	

MAIL DATE	DELIVERY MODE
02/07/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

mn

Office Action Summary	Application No.	Applicant(s)	
	10/601,444	BRUNDAGE ET AL.	
	Examiner	Art Unit	
	Dangelino N. Gortayo	2168	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *31 October 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-6 and 8-23* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-6 and 8-23* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 10/31/2007 has been entered.

Response to Amendment

2. In the amendment filed on 10/31/07, claims 1, 11, 17, and 21 have been amended. The currently pending claims considered below are Claims 1-6 and 8-23.

Claim Rejections - 35 USC § 112

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claims 11 and 21 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Claim 11 recites the limitation "the semantic representation" in line 14. There is insufficient antecedent basis for this limitation in the claim. For purposes of examination, examiner assumes the semantic representation

refers to the previously disclosed “intermediate language representation” disclosed in the claims.

Claim Rejections - 35 USC § 101

5. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

6. Claim 17 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. Claim 17 recites the limitation ”A computer readable medium having computer-executable instructions” starting in line 1 of the claim, which is intangible and non-statutory as disclosed by the specification in block [0036], disclosed to be a computer system containing computer readable media in lines 1-2. The computer readable medium is disclosed to be part of a computer network comprising program instructions sent over “modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal... communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media.” (block 0036 lines 21-34 of the instant application). This renders the claims as program per se. Appropriate correction is required.

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claims 1-6 and 8-23 are rejected under 35 U.S.C. 103(a) as being anticipated by <u>Manikutty</u> et al. (US Patent 7,120,645 B2) in view of <u>Shanmugasundaram</u> et al. ("Relational Databases for Querying XML Documents: Limitations and Opportunities", Published Sept. 1999 in "Proceedings of the Twenty-Fifth International Conference on Very Large Data Bases", Pages 302-314)

As per claim 1, <u>Manikutty</u> teaches ""A method for semantic representation of one or more XML language inquiries across relational and non-relational data sources" (see Abstract)

"receiving at least one inquiry" (Figure 3 reference 310, column 10 lines 25-33, column 16 lines 56-58, wherein a query is received);

"defining a plurality of nodes of a graph structure which represents the at least one inquiry, the graph structure having at least one node object for every operation within the at least one received inquiry" (Figure 5, column 1 lines 61-67, column 14 lines 55-66. column 16 lines 18-26, column 20 lines 63-67, column 21 lines 13-30, wherein a set of XML generation operations and rules to convert between XML operations to

canonical operations are established, wherein canonical operations are represented as nodes in a normalized tree of canonical functions);

"translating each of the at least one node objects using operators" (column 16 line 59 – column 17 line 8, column 18 lines 37-43, lines 57-65, column 19 lines 57-62, column 20 lines 26-42, wherein a query is translated based on certain conditions and generation rules);

"generating a semantic representation having the graph structure wherein the semantic representation explicitly describes a meaning of the one or more XML language inquiries" (Figure 5, column 20 line 63 – column 21 line 44, column 22 lines 47-67, wherein queries are translated to a normalized tree of canonical functions, wherein the nodes of a tree represent operations, and the tree describing the query) "and wherein the semantic representation decouples front-end language compilers from back-end query engines that use the semantic representation" (column 21 line 48 – column 22 line 47, wherein a second, more simplified tree from the first normalized tree decouples SQL operations and XML operations).

Manikutty does not teach "the semantic representation including a tuple operation having three child nodes, the child nodes comprising a list of iterators that construct tuple space, a clause that filters the tuple space, and a clause that produces an outcome of the tuple space, wherein each iterator in the semantic representation of a tuple node corresponds to one column in the tuple space"

Shanmugasundaram teaches "the semantic representation including a tuple operation having three child nodes, the child nodes comprising a list of iterators that

construct tuple space, a clause that filters the tuple space, and a clause that produces an outcome of the tuple space, wherein each iterator in the semantic representation of a tuple node corresponds to one column in the tuple space" (page 4 section 3.1 "Simplifying DTDs", page 5 Figure 8 and 9, page 5 section 3.3 "The Basic Inlining Technique", page 10 section 5. "Converting Relational Results to XML", wherein conversion of XML queries involves translation to a representation of relational tuples, including tag variables, grouping of data, and heterogeneous results).

It would have been obvious at the time of the invention for one of ordinary skill in the art to combine Manikutty's method of translating a plurality of queries to a representation containing a graph structure with Shanmugasundaram's method of translating queries to a representation containing relational tuples, including a method of construction of tuple space, a method to limit the range of tuples, and a method to output data from relational tuples. This gives the user the ability to utilize tuple information in a graph structure that represents a plurality of queries from a user, which allows a user to query multiple heterogeneous data sources utilizing methods to access a relational database. The motivation for doing so would be to access XML data and queries in a similar fashion to relational data (pages 2 paragraph 4).

As per claim 2, Manikutty teaches "the semantic representation is an intermediate language representation formed for interpretation and execution by a target query engine" (column 10 lines 53-60)

As per claim 3, Manikutty teaches "wherein the non-relational data sources comprise one or more of a text document, a spreadsheet, and a non-relational database" (column 8 lines 58-66)

As per claim 4, Manikutty teaches "the generating step further comprises breaking down high level operations of the received inquiry into explicit parts" (column 14 lines 45-54).

As per claim 5, Manikutty teaches "the explicit parts are common across multiple XML languages" (column 5 lines 52-62, column 6 lines 10-23).

As per claim 6, Manikutty teaches "the operators comprise one or more of special operators, data sources, literals, Boolean operators, sequence operators, arithmetic operators, string operators, value comparison operators, node comparison operators, tuple spaces, function definition and invocation, XML navigation, XML construction, XML property accessors, type operators, language specific operators, and data manipulation operators" (Table 12, column 15 lines 21-25, column 26 line 64 – column 31 line 62).

As per claim 8, Manikutty teaches "at least one received inquiry comprises one or more of an XML query language and an XML view definition language" (column 10 lines 25-33, column 16 lines 56-58).

As per claim 9, Manikutty teaches "the at least one received inquiry comprises one or more of an XPath, an XSLT, an XQuery, a DML, an OPath, and an Annotated Schema inquiry." (column 5 line 63 – column 6 line 9, column 6 lines 23-36).

As per claim 10, Manikutty teaches "the semantic language representation allows XML queries over XML views of relational data" (column 5 lines 36-50, column 17 lines 18-25).

As per claim 11, Manikutty teaches "semantics interpreter for expressing a meaning of one or more of an XML query and an XML view across multiple data source" (see Abstract);

"an input for receiving the one or more of an XML query and an XML view which form an inquiry" (Figure 3 reference 310, column 10 lines 25-33, column 16 lines 56-58, wherein a query is received);

"a graph structure generator for defining node objects for every operation within the inquiry" (Figure 5, column 1 lines 61-67, column 14 lines 55-66. column 16 lines 18-26, column 20 lines 63-67, column 21 lines 13-30, wherein a set of XML generation operations and rules to convert between XML operations to canonical operations are established, wherein canonical operations are represented as nodes in a normalized tree of canonical functions);

"a translator for assigning operators for each node object wherein the operators break down operations of the inquiry into explicit parts" (column 16 line 59 – column 17 line 8, column 18 lines 37-43, lines 57-65, column 19 lines 57-62, column 20 lines 26-42, wherein a query is translated based on certain conditions and generation rules);

"output for providing the explicit parts as an intermediate language representation for expressing the meaning of the one or more of an XML query and an XML view"

(Figure 5, column 20 line 63 – column 21 line 44, column 22 lines 47-67, column 21 line 48 – column 22 line 47, wherein queries are translated to a normalized tree of canonical functions, wherein the nodes of a tree represent operations, and the tree describing the query)

"wherein the intermediate language representation decouples front-end language compilers from back-end query engines that use the intermediate language representation" (column 21 line 48 – column 22 line 47, wherein a second, more simplified tree from the first normalized tree decouples SQL operations and XML operations) "such that utilization of the intermediate language representation in the semantics interpreter, when used in a compiler system having M front-front-end languages and N back-end search engines, has a complexity of M plus N compiler implementations" (column 4 lines 36-62, column 9 line 59 - column 10 line 24, column 21 line 48 - column 22 line 29, wherein the results of the XML generating sub-query are expanded into a second tree of XML operations, the front end language being XML and the back end being XPath traversals)

Manikutty does not teach "the intermediate language representation including a tuple operation having three child nodes, the child nodes comprising a list of iterators that construct tuple space, a clause that filters the tuple space, and a clause that produces an outcome of the tuple space, wherein each iterator in the semantic representation of a tuple node corresponds to one column in the tuple space"

Shanmugasundaram teaches "the intermediate language representation including a tuple operation having three child nodes, the child nodes comprising a list of

iterators that construct tuple space, a clause that filters the tuple space, and a clause that produces an outcome of the tuple space, wherein each iterator in the semantic representation of a tuple node corresponds to one column in the tuple space" (page 4 section 3.1 "Simplifying DTDs", page 5 Figure 8 and 9, page 5 section 3.3 "The Basic Inlining Technique", page 10 section 5. "Converting Relational Results to XML", wherein conversion of XML queries involves translation to a representation of relational tuples, including tag variables, grouping of data, and heterogeneous results).

It would have been obvious at the time of the invention for one of ordinary skill in the art to combine Manikutty's method of translating a plurality of queries to a representation containing a graph structure with Shanmugasundaram's method of translating queries to a representation containing relational tuples, including a method of construction of tuple space, a method to limit the range of tuples, and a method to output data from relational tuples. This gives the user the ability to utilize tuple information in a graph structure that represents a plurality of queries from a user, which allows a user to query multiple heterogeneous data sources utilizing methods to access a relational database. The motivation for doing so would be to access XML data and queries in a similar fashion to relational data (pages 2 paragraph 4).

As per claim 12, Manikutty teaches "the multiple data sources comprise relational and non-relational data sources" (column 5 lines 26-35, column 8 lines 58-66)

As per claim 13, this claim is rejected on grounds corresponding to the arguments given above for rejected claim 3 and is similarly rejected.

As per claim 14, this claim is rejected on grounds corresponding to the arguments given above for rejected claim 6 and is similarly rejected.

As per claim 15, this claim is rejected on grounds corresponding to the arguments given above for rejected claim 5 and is similarly rejected.

As per claim 16, Manikutty teaches "the intermediate language representation is formed for interpretation and execution by a target query engine" (column 10 lines 53-60)

As per claim 17, Manikutty teaches "A computer-readable medium having computer-executable instructions for performing a method of intermediate language representation of a received inquiry" (see Abstract);

"receiving one or more of an XML query and an XML view forming the received inquiry" (Figure 3 reference 310, column 10 lines 25-33, column 16 lines 56-58, wherein a query is received);

"defining node objects for every operation within the received inquiry" (column 1 lines 61-67, column 14 lines 55-66. column 16 lines 18-26, column 20 lines 63-67, wherein a set of XML generation operations and rules to convert between XML operations to canonical operations are established, wherein canonical operations are represented as nodes in a normalized tree of canonical functions);

"translating each node using operators which break down operations of the received inquiry into explicit parts" (column 16 line 59 – column 17 line 8, column 18

lines 37-43, lines 57-65, column 19 lines 57-62, column 20 lines 26-42, wherein a query is translated based on certain conditions and generation rules);

"generating instructions corresponding to the explicit parts forming an intermediate language representation for subsequent queries over one or more of relational and non-relational data sources,"(Figure 5, column 20 line 63 – column 21 line 44, column 22 lines 47-67, wherein queries are translated to a normalized tree of canonical functions, wherein the nodes of a tree represent operations, and the tree describing the query) "wherein the intermediate language representation decouples front-end language compilers from back-end query engines that use the semantic representation" (column 21 line 48 – column 22 line 47, wherein a second, more simplified tree from the first normalized tree decouples SQL operations and XML operations).

"such that utilization of the intermediate language representation in the semantics interpreter, when used in a compiler system having M front-front-end languages and N back-end search engines, has a complexity of M plus N compiler implementations" (column 4 lines 36-62, column 9 line 59 - column 10 line 24, column 21 line 48 - column 22 line 29, wherein the results of the XML generating sub-query are expanded into a second tree of XML operations, the front end language being XML and the back end being XPath traversals)

Manikutty does not teach "wherein the intermediate language representation comprises an explicit description of a meaning on the received inquiry"

Shanmugasundaram teaches wherein the intermediate language representation comprises an explicit description of a meaning on the received inquiry" (page 4 section 3.1 "Simplifying DTDs", page 5 Figure 8 and 9, page 5 section 3.3 "The Basic Inlining Technique", page 10 section 5. "Converting Relational Results to XML", wherein conversion of XML queries involves translation to a representation of relational tuples, including tag variables, grouping of data, and heterogeneous results).

It would have been obvious at the time of the invention for one of ordinary skill in the art to combine Manikutty's method of translating a plurality of queries to a representation containing a graph structure with Shanmugasundaram's method of translating queries to a representation containing relational tuples, including a method of construction of tuple space, a method to limit the range of tuples, and a method to output data from relational tuples. This gives the user the ability to utilize tuple information in a graph structure that represents a plurality of queries from a user, which allows a user to query multiple heterogeneous data sources utilizing methods to access a relational database. The motivation for doing so would be to access XML data and queries in a similar fashion to relational data (pages 2 paragraph 4).

As per claim 18, this claim is rejected on grounds corresponding to the arguments given above for rejected claim 6 and is similarly rejected.

As per claim 19, this claim is rejected on grounds corresponding to the arguments given above for rejected claim 5 and is similarly rejected.

As per claim 20, this claim is rejected on grounds corresponding to the arguments given above for rejected claim 8 and is similarly rejected.

As per claim 21, Manikutty teaches "A computer system for generating a semantic representation of an inquiry" (see Abstract)

"a processor for executing computer instructions and at least one module" (Figure 6 reference 604 and column 25 lines 27-42)

"an input function for receiving one or more of an XML query and an XML view which forms the inquiry" (Figure 3 reference 310, column 10 lines 25-33, column 16 lines 56-58, wherein a query is received);

"a graph structure generator for defining node objects for every operation within the inquiry" (Figure 5, column 1 lines 61-67, column 14 lines 55-66. column 16 lines 18-26, column 20 lines 63-67, column 21 lines 13-30, wherein a set of XML generation operations and rules to convert between XML operations to canonical operations are established, wherein canonical operations are represented as nodes in a normalized tree of canonical functions);

"a translator function for assigning operators for each node object wherein the operators break down operations of the inquiry into explicit parts" (column 16 line 59 – column 17 line 8, column 18 lines 37-43, lines 57-65, column 19 lines 57-62, column 20 lines 26-42, wherein a query is translated based on certain conditions and generation rules);

"an output for providing the explicit parts as an intermediate language representation for expressing a meaning of the XML query and the XML view" (Figure 5, column 20 line 63 – column 21 line 44, column 22 lines 47-67, wherein queries are

translated to a normalized tree of canonical functions, wherein the nodes of a tree represent operations, and the tree describing the query) "wherein the intermediate language representation decouples front-end language compilers from back-end query engines that use the semantic representation" (column 21 line 48 – column 22 line 47, wherein a second, more simplified tree from the first normalized tree decouples SQL operations and XML operations).

"such that utilization of the intermediate language representation in the semantics interpreter, when used in a compiler system having M front-front-end languages and N back-end search engines, has a complexity of M plus N compiler implementations" (column 4 lines 36-62, column 9 line 59 - column 10 line 24, column 21 line 48 - column 22 line 29, wherein the results of the XML generating sub-query are expanded into a second tree of XML operations, the front end language being XML and the back end being XPath traversals)

"wherein the at least one module comprises one or more of one or more software modules and one or more hardware modules" (column 25 lines 1-42)

"wherein the intermediate language representation is executed directly by one execution engine, and is translated to SQL before execution by a second execution engine, and is executed partially in a third execution engine wherein a balance of the intermediate language representation is executed in a fourth execution engine." (Figure 3 references 320, 330, 340, 350, 360, and column 16 line 8 – column 17 line 33, wherein the XML operations are replaced by SQL operations, the queries are rewritten,

an SQL processor evaluates the SQL operations , and the XML operations are evaluated by an SQL/XML processor)

Manikutty does not teach "the intermediate language representation including a tuple operation having three child nodes, the child nodes comprising a list of iterators that construct tuple space, a clause that filters the tuple space, and a clause that produces an outcome of the tuple space, wherein each iterator in the semantic representation of a tuple node corresponds to one column in the tuple space"

Shanmugasundaram teaches "the intermediate language representation including a tuple operation having three child nodes, the child nodes comprising a list of iterators that construct tuple space, a clause that filters the tuple space, and a clause that produces an outcome of the tuple space, wherein each iterator in the semantic representation of a tuple node corresponds to one column in the tuple space" (page 4 section 3.1 "Simplifying DTDs", page 5 Figure 8 and 9, page 5 section 3.3 "The Basic Inlining Technique", page 10 section 5. "Converting Relational Results to XML", wherein conversion of XML queries involves translation to a representation of relational tuples, including tag variables, grouping of data, and heterogeneous results).

It would have been obvious at the time of the invention for one of ordinary skill in the art to combine Manikutty's method of translating a plurality of queries to a representation containing a graph structure with Shanmugasundaram's method of translating queries to a representation containing relational tuples, including a method of construction of tuple space, a method to limit the range of tuples, and a method to output data from relational tuples. This gives the user the ability to utilize tuple

information in a graph structure that represents a plurality of queries from a user, which allows a user to query multiple heterogeneous data sources utilizing methods to access a relational database. The motivation for doing so would be to access XML data and queries in a similar fashion to relational data (pages 2 paragraph 4).

As per claim 22, this claim is rejected on grounds corresponding to the arguments given above for rejected claim 6 and is similarly rejected.

As per claim 23, this claim is rejected on grounds corresponding to the arguments given above for rejected claim 5 and is similarly rejected.

Response to Arguments

9. Applicant's arguments with respect to the rejection for claim 1-6 and 8-23 under 35 USC 102(e) have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

10. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Marron et al. (US Patent 6,901,410 B2)

Christianson et al. (US Patent 7,124,144 B2)

Petropoulos et al. (US Patent 7,203,678 B1)

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Dangelino N. Gortayo whose telephone number is (571)272-7204. The examiner can normally be reached on M-F 7:30-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tim T. Vo can be reached on (571)272-3642. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Dangelino N. Gortayo
Examiner

Tim T. Vo
SPE

TIM VO
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100